LP-Based Identification of general nonlinear multivariable maps: Preliminary results

Mazen Alamir

LP-Based Identification of general nonlinear multivariable maps: Preliminary results

Mazen Alamir

Abstract

In this paper, a systematic way to approximate general nonlinear maps by means of linear programs (LP) is proposed. The argument of the map to be identified is a parameter of the LP problem while the LP-matrices are computed based on a learning data. Some examples are given to illustrate the proposed algorithm. Note that this is a working document for which several versions will be available. The author would be grateful for any constructive comments on his proposal.

Index Terms

Nonlinear identification, Linear Programming.

I. Introduction

Identification of general functions is a key issue in many engineering paradigms such as control design, state/parameter estimation and diagnosis among many others. The solutions to this problem are generally classified based on the structure of the model (linear, Wiener, neural networks and so on). Once this structure is fixed, the identification problem reduces to find the optimal set of parameters that minimizes some criterion (L_2, L_1, etc) and given some assumptions on the noise that affects the data used for the identification.

Some of the candidate structures to define the identification problem are universal in the sense that, at least theoretically, by increasing the number of parameters, it is possible to fit the learning data up to any desired precision. Neural networks, multivariate polynomials or piecewise affine functions are examples of such universal structures while affine models are not universal.

A common idea regarding identification is the one stating that the resulting model can be explicitly given through mathematical expressions in which the identified parameters appear explicitly and the value of the identified function can then be computed by a sequence of explicit expressions evaluation.

With the increasing power of nowadays computers, solving LP problems can be done so fast and so accurately that it might be viewed as an *explicit* expression. Now since the solution of an LP problem is precisely a piece-wise affine function of the decision variable, it comes that LP structure is a candidate parametrization of the universal class of piecewise affine structure.

This paper exploits this property by proposing a systematic algorithm to compute LP parameters so that the solution of the resulting LP problem matches the function to be identified to any desired precision.

II. Definitions, Notation and Preliminary Results

Consider a function to be identified that is given by:

$$y = F(x) \quad \text{where } (x, y) \in \mathbb{R}^n \times \mathbb{R}^m \tag{1}$$

we assume that we dispose of a learning data D such that:

$$\mathcal{D} := \left\{ x^{(\ell)}, y^{(\ell)} \right\}_{\ell=1}^{n_d} \quad ; \quad y^{(\ell)} = F(x^{(\ell)}) \tag{2}$$

The aim of this paper is to propose an algorithm that enables to compute the x-dependent parameters of a linear program:

$$\begin{array}{rlll} \mathcal{P}(x) \ : & \min\limits_{y,\epsilon} & \mathbf{1}^T \epsilon & \\ & \text{under} & \epsilon \geq 0 & \\ & \text{and} & |y_i - \phi_i(x)| \leq \epsilon_i & i \in \{1, \ldots, m\} \\ & \text{and} & Ay \leq B(x) & \end{array} \tag{3}$$

M. Alamir is with CNRS/Gipsa-lab, Control Systems Dept., University of Grenoble, France e-mail: mazen.alamir@grenoble-inp.fr (http://www.mazenalamir.fr).

with

$$A \in \mathbb{R}^{n_a \times m}, B \in \mathbb{R}^{n_a} \tag{4}$$

such that the solution $y^*(x)$ tightly approximates $F(x)$:

$$y^*(x) \approx F(x) \tag{5}$$

for all x represented in the identification data $\mathcal{D}$.

In the formulation above, the map $\phi : \mathbb{R}^n \to \mathbb{R}^m$ is given once for all and is not a part of what is to be computed. As it is shown later on, the choice of $\phi(x)$ is problem dependent although trivial choice can always be used.

The following specific structure is used in the sequel for $B(x)$:

$$B(x) := \sum_{j=1}^{n_b} B_j \cdot \psi_j(x) \tag{6}$$

where:

$$B_j \in \mathbb{R}^{n_a \times 1} \quad ; \quad \psi_j(x) \in \mathbb{R} \tag{7}$$

Here again, the choice of the functions $\psi_j(x)$ is problem-dependent. A good choice can reduce the number n_a of constraints that would be necessary to achieve the desired precision. A possible set of functions ψ_j can be defined by:

$$n_b = 2 \quad ; \quad \psi_1(x) = \mathbf{1} \in \mathbb{R}^n \quad ; \quad \psi_2(x) = x \tag{8}$$

Once $\phi(\cdot)$ and ψ_j, $j = 1, \ldots, n_b$ are chosen, the parameters of the solution are the matrices A, B_j, $j = 1, \ldots, n_b$. This gathers a number n_p of parameters equal to:

$$n_p = n_a \times (m + n_b) \tag{9}$$

These parameters are gathered in a single column vector $p \in \mathbb{R}^{n_p}$ that is used in the sequel as the parameter to be computed by the proposed algorithm.

In the sequel, the matrices A, B_1,...,B_{n_b} corresponding to some value of the parameter p are denoted by $A(p)$, $B_1(p)$,...,$B_{n_b}(p)$.

Now, since the definition of the LP problem $\mathcal{P}(x)$ depends on the value of the parameter p, the following notation is used in order to explicitly underline this dependence:

$$\mathcal{P}(x) \quad \text{becomes} \quad \mathcal{P}(x, p) \tag{10}$$

Assuming that p is defined by a column-based extraction of the matrix:

$$\begin{pmatrix} A & B_1 & \ldots & B_{n_b} \end{pmatrix} \in \mathbb{R}^{n_a \times (m + n \cdot n_b)} \tag{11}$$

it comes that the inequality $Ay \leq B(x)$ can be written using p as follows:

$$[M(x, y)]\, p \leq 0 \tag{12}$$

where the matrix $M(x, y)$ is given by

$$M(x, y) = \begin{pmatrix} y^T & -\psi_1(x) & \ldots & -\psi_{n_b}(x) \end{pmatrix} \otimes \mathbb{I}_{n_a} \tag{13}$$

Using this definition, the following matrix can be defined that involves the whole learning set $\mathcal{D}$:

$$\bar{M}(\mathcal{D}) := \begin{pmatrix} M(x^{(1)}, y^{(1)}) \\ \vdots \\ M(x^{(n_d)}, y^{(n_d)}) \end{pmatrix} \in \mathbb{R}^{(n_a \cdot n_d) \times n_p} \tag{14}$$

so that the satisfaction of the constraint:

$$\left[\bar{M}(\mathcal{D})\right] p \leq 0 \in \mathbb{R}^{(n_a \cdot n_d) \times 1} \tag{15}$$

is equivalent to the satisfaction of the constraints:

$$Ay^{(\ell)} \leq B(x^{(\ell)}) \quad \forall \ell \in \{1, \ldots, n_d\} \tag{16}$$

This can be summarized in the following Lemma:

Lemma 2.1: If p satisfies (15) then for all $\ell \in \{1, \ldots, n_d\}$, the pair $(y^{(\ell)}, \epsilon^{(\ell)})$ defined by:

$$y^{(\ell)} = F(x^{(\ell)}) \quad ; \quad \epsilon_i^{(\ell)} = |y_i^{(\ell)} - \phi_i(x^{(\ell)})| \tag{17}$$

is an admissible solution to the linear programming problem $\mathcal{P}(x^{(\ell)}, p)$. (The corresponding costs will be denoted in the sequel by $J^{(\ell)}$). $\diamondsuit$

PROOF. This comes from the fact that by construction, the constraint (15) is equivalent to the satisfaction of the constraint $Ay^{(\ell)} \leq B(x^{(\ell)})$, the other two constraints are satisfied by the very definition of $\epsilon_i^{(\ell)}$ given by (17). $\square$

This lemma proves that when one solves the optimization problem $\mathcal{P}(x^{(\ell)}, p^{(0)})$ where $p^{(0)}$ is the minimum norm solution of (15):

$$\begin{array}{rl} p^{(0)} & \leftarrow \underset{p}{\text{argmin}} \|p\|^2 \\ & \text{under} \quad \left[\bar{M}(\mathcal{D})\right] p \leq 0 \end{array} \tag{18}$$

then, the ideal value one would like to obtain (namely $y^{(\ell)} = F(x^{(\ell)})$) is admissible. However, the resulting optimal solution, denoted by $\hat{y}^{(\ell,0)}$ is not necessarily equal to $y^{(\ell)}$ and the optimal cost achieved by $\hat{y}^{(\ell,0)}$, denoted hereafter by $\hat{J}^{(\ell,0)}$ is by definition lower than $J^{(\ell)}$:

$$(\forall \ell \in \{1, \ldots, n_d\}) \quad \hat{J}^{(\ell,0)} \leq J^{(\ell)} \tag{19}$$

In order to properly introduce the iterative process, the following notation is used:

$$M^{(0)} = \bar{M}(\mathcal{D}) \quad ; \quad S^{(0)} = 0 \in \mathbb{R}^{n_a} \tag{20}$$

so that (18) can be rewritten as follows:

$$\begin{array}{rll} \mathcal{Q}^{(0)}: & p^{(0)} & \leftarrow \underset{p}{\text{argmin}} \|p\|^2 \\ & & \text{under} \quad \left[\bar{M}^{(0)}\right] p \leq S^{(0)} \end{array} \tag{21}$$

Let us denote by $\ell^{(0)}$ the index of the worst point, namely:

$$\ell^{(0)} \leftarrow \arg \max_{\ell \in \{1,\ldots,n_d\}} |\hat{J}^{(\ell,0)} - J^{(\ell)}| \tag{22}$$

The idea is to add a new constraint such that the solution $y^{(\ell^{(0)},0)}$ is no more admissible for the new future problem at $\mathcal{P}(x^{(\ell^{(0)})}, p^{(1)})$. In order to do so, remember that $\hat{y}^{(\ell^{(0)},0)}$ is admissible for $\mathcal{P}(x^{(\ell^{(0)})}, p^{(0)})$ which means that:

$$\left[M(x^{(\ell^{(0)})}, \hat{y}^{(\ell^{(0)},0)})\right] p^{(0)} \leq 0 \in \mathbb{R}^{n_a} \tag{23}$$

Let $i^{(0)}$ be the index that corresponds to the maximum component of the l.h.s of (23), namely:

$$i^{(0)} \leftarrow \arg \max_{i \in \{1,\ldots,n_a\}} \left[M_i(x^{(\ell^{(0)})}, \hat{y}^{(\ell^{(0)},0)})\right] p^{(0)} \tag{24}$$

This is the component that is the closest to 0 and therefore, a possible new constraint that can be added to exclude $\hat{y}^{(\ell^{(0)},0)}$ from the possible solution for $\mathcal{P}^{(1)}(x^{(\ell^{(0)})})$ is the scalar constraint given by

$$\left[E^{(0)}\right] p \geq \eta > 0 \tag{25}$$

where $E^{(0)}$ is defined by:

$$E^{(0)} := \left[M_{i^{(0)}}(x^{(\ell^{(0)})}, \hat{y}^{(\ell^{(0)},0)})\right] \tag{26}$$

while $\eta > 0$ is a strictly positive real.

Using $E^{(0)}$ and η, the quadratic optimization problem (21) is replaced by the the new problem $\mathcal{Q}^{(1)}$ defined by:

$$\begin{array}{rll} \mathcal{Q}^{(1)}: & p^{(1)} & \leftarrow \underset{p}{\text{argmin}} \|p\|^2 \\ & & \text{under} \quad \left[\bar{M}^{(1)}\right] p \leq S^{(1)} \end{array} \tag{27}$$

where:

$$\bar{M}^{(1)} \leftarrow \begin{pmatrix} \bar{M}^{(0)} \\ -E^{(0)} \end{pmatrix} \quad ; \quad S^{(1)} \leftarrow \begin{pmatrix} S^{(0)} \\ -\eta \end{pmatrix} \tag{28}$$

By solving this new optimization problem, a new parameter $p^{(1)}$ can be computed and the process can be repeated through the successive computation of $\hat{y}^{(\ell,1)}$, $\ell^{(1)}$, $i^{(1)}$, $E^{(1)}$. This enables the new optimization problem $\mathcal{Q}^{(2)}$ to be define that leads to

the new iterate $p^{(2)}$ and so on. The choice of the parameter η used in the new constraint (25) and leading to the definition of the new Quadratic Program $\mathcal{Q}^{(1)}$ invoked in (29) needs to be more clearly defined. Indeed, higher value of η accelerates the convergence as it leads to the exclusion of larger region in a single step. However, too large values may lead to infeasible QPs.

That is the reason why for all $i \geq 0$, the QP problem $\mathcal{Q}^{(i+1)}$ is redefined as follows:

$$\begin{array}{lll} \mathcal{Q}^{(i+1)}: & (p^{(i+1)}, \eta^{(i+1)}) \leftarrow \underset{(p,\eta)}{\text{argmin}} \left[\|p\|^2 - r \times \eta\right] \\ & \text{under} \quad \left[\bar{M}^{(i+1)}\right] p \leq S^{(i+1)}(\eta) \\ & \text{and} \quad \eta \in [0, \bar{\eta}] \end{array} \tag{29}$$

where:

$$\bar{M}^{(i+1)} \leftarrow \begin{pmatrix} \bar{M}^{(i)} \\ -E^{(i)} \end{pmatrix} \quad ; \quad S^{(i+1)} \leftarrow \begin{pmatrix} S^{(i)} \\ -\eta \end{pmatrix} \tag{30}$$

This leads to the algorithm proposed in the next section.

III. The proposed Algorithm

Note that the input of the algorithm are:

1) The learning data $\mathcal{D}$ defined by (2),
2) The number of constraints n_a,
3) The number n_b of functions used in (6),
4) The function $\phi : \mathbb{R}^n \to \mathbb{R}^n$,
5) The functions $\psi_j : \mathbb{R}^n \to \mathbb{R}^n$,
6) The stopping condition threshold $\varepsilon > 0$

Algorithm 1 $[A, B_1, \ldots, B_{n_b}] = \mathcal{A}(\mathcal{D}, n_a, n_b, \phi, \{\psi_j\}_{j=1}^{n_b}, \bar{\eta})$

Initialization. $M^{(0)} \leftarrow \bar{M}(\mathcal{D})$, $S^{(0)} = 0 \in \mathbb{R}^{n_a}$, $i = 0$
again $\leftarrow$ true.
while ((again) and ($i \leq i_{max}$)) **do**
  $(p^{(i)}, \eta^{(i)}, \text{flag}) \leftarrow$ Solve $\mathcal{Q}^{(i)}$ defined by $(\bar{M}^{(i)}, S^{(i)})$
  if (flag=success) **then**
    for $i = \ell, \ldots, n_d$ **do**,
      $[\hat{y}^{(\ell,i)}, \hat{J}^{(\ell)}] \leftarrow$ Solve $\mathcal{P}(x^{(\ell)}, p^{(i)})$
    end for
    $l^* \leftarrow \arg \max_{\ell=1,\ldots,n_d} |\hat{J}^{(\ell,i)} - J^{(\ell)}|$
    $i^* \leftarrow \arg \max_{i=1,\ldots,n_a} \left[M_i(x^{(\ell^*)}, \hat{y}^{(\ell^*,i)})p^{(i)}\right]$
    $E^{(i)} \leftarrow M_{i^*}(x^{(\ell^*)}, \hat{y}^{(\ell^*,i)})$
    $\bar{M}^{(i+1)} \leftarrow \begin{bmatrix} \bar{M}^{(i)} \\ -E^{(i)} \end{bmatrix}$, $S^{(i+1)} \leftarrow \begin{bmatrix} S^{(i)} \\ -\eta^{(i)} \end{bmatrix}$
    $p \leftarrow p^{(i)}$, $i \leftarrow i + 1$
  else
    again $\leftarrow$ false.
  end if
end while
$A \leftarrow A(p)$, $B_j \leftarrow B_j(p)$, $j = 1, \ldots, n_b$

IV. Illustrative Examples

A. Example 1

Consider the following relationship:

$$y = F(x) = \min\left\{x_2 - x_1^2 + 4, 3(x_1 - 1)^5 + 0.1\left[\cos(x_1)\right]^2\right\} \tag{31}$$

that we would like to approximate using x-dependent LP structure over the interval of values:

$$x \in [-1, 3] \times [1, 6] \tag{32}$$

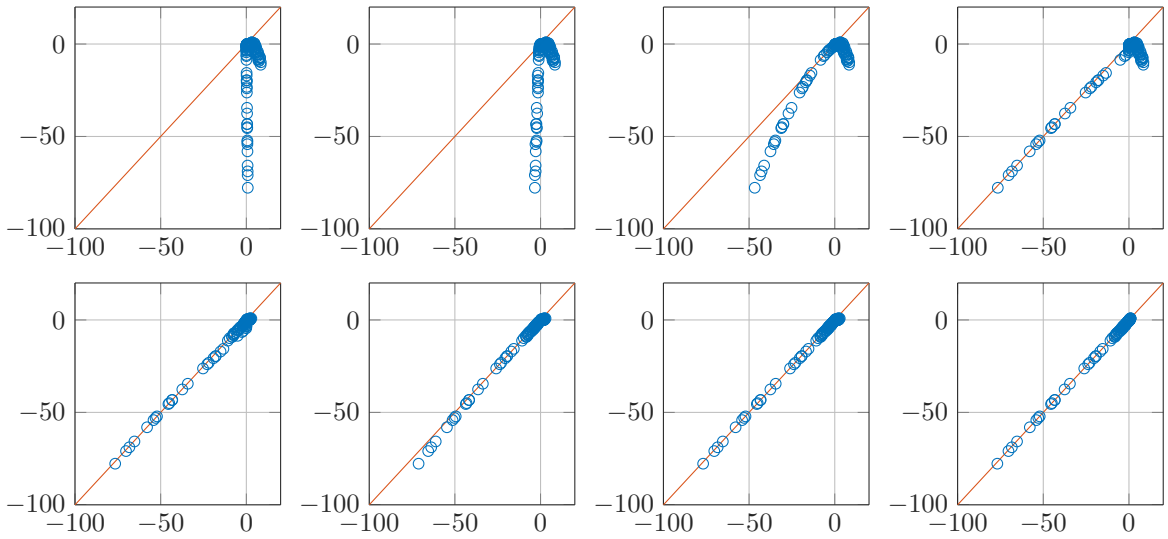

Fig. 1. Identification results for the first 8 iteration of the proposed algorithm. The abscissas represent the value of the predicted values $\hat{y}^{(\ell,i)}$, $\ell = 1, \ldots, n_d$ while the vertical coordinate is the true value of the function given by (31)

Note that this interval does enhance the nonlinearities as well as the saturation of the map.

The learning data set if generated randomly on the above interval. The following parameters and function definitions have been used to perform the identification of the x-dependent LP:

$$n_a = 3 \ ; \ r = 1000 \ ; \ n_d = 100 \ ; \ \bar{\eta} = 10^2 \ ; \ \phi(x) = x_1^2 \ ; \ \psi_j(x) \in \{1, x_1, x_2, x_1^2, x_1^3\} \tag{33}$$

Figure 1 shows the identification results for the first 8 iteration of the proposed algorithm. The abscissas represent the value of the predicted values $\hat{y}^{(\ell,i)}$, $\ell = 1, \ldots, n_d$ while the vertical coordinate is the true value of the function given by (31).

The identification leads to the following matrices A and B invoked in the general form of the targeted x-dependent LP structure $\mathcal{P}(x)$ (35):

$$A := \begin{pmatrix} 4.1508 \\ 5.2172 \\ 56.8916 \end{pmatrix} \quad ; \quad B := \begin{pmatrix} -12.0611 & 61.1928 & -0.0004 & -89.3475 & 227.0864 \\ 1.6826 & 2.1396 & 4.2485 & 1.5429 & -3.4573 \\ 4.6195 & -0.4127 & -1.1362 & -0.4585 & 9.9353 \end{pmatrix} \tag{34}$$

So that for any given x, the LP-based identification of y is given by te solution of the following LP:

$$\begin{array}{lll} \mathcal{P}(x) \ : & \min\limits_{y,\epsilon} & \mathbf{1}^T \epsilon \\ & \text{under} & \epsilon \geq 0 \\ & \text{and} & |y - x_1^2| \leq \epsilon \\ & \text{and} & \begin{pmatrix} 4.1508 \\ 5.2172 \\ 56.8916 \end{pmatrix} y \leq \begin{pmatrix} -12.0611 & 61.1928 & -0.0004 & -89.3475 & 227.0864 \\ 1.6826 & 2.1396 & 4.2485 & 1.5429 & -3.4573 \\ 4.6195 & -0.4127 & -1.1362 & -0.4585 & 9.9353 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1^3 \end{pmatrix} \end{array} \tag{35}$$

B. Example 2

In the second example, the proposed method is applied to identify an LP model for the maximum power that can be extracted from wind/tidal power turbine. This maximum power takes the form depicted in Figure 2 The algorithm proposed in this work has been used on this example using the following parameters:

$$n_d = 50 \quad ; \quad n_a = 2 \quad ; \quad \phi(v) = 500 \quad ; \quad \psi_j(v) \in \{1, v, v^3, v^5\} \tag{36}$$

the behavior if the algorithm during the successive iterations is shown on Figure 3. This results in a v-dependent LP with two linear constraints and 1 positivity constraint where the matrices are given by:

$$A := \begin{pmatrix} 2.0320 \\ 1.1103 \end{pmatrix} \quad ; \quad B := \begin{pmatrix} 2.5453 & -4.7451 & -4.3585 & 15.8110 \\ 0.4072 & 3.1297 & 21.2405 & 1.5614 \end{pmatrix} \tag{37}$$

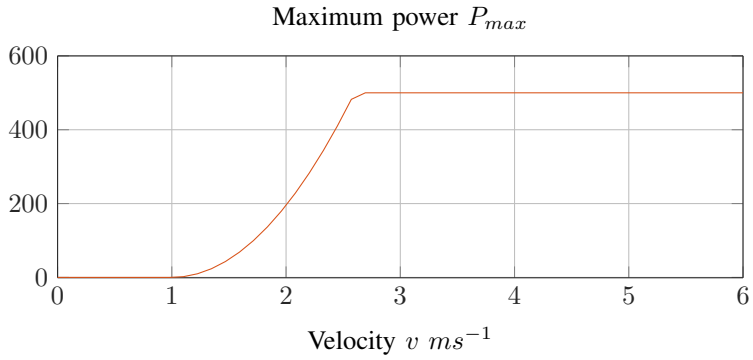

Fig. 2. Typical maximum power form as a function of the velocity of wind and/or tidal currents.

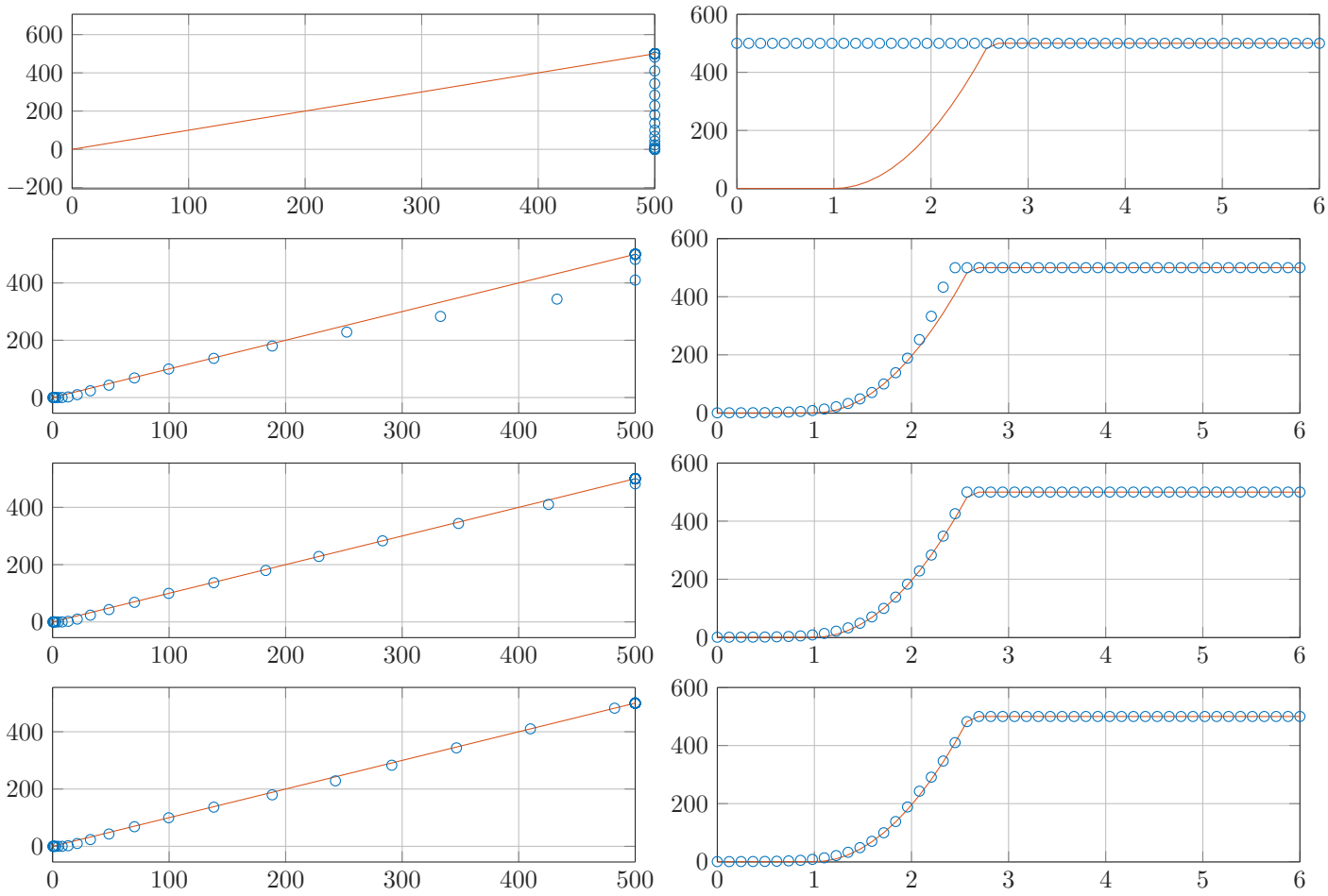

Fig. 3. Evolution of the iteration for the maximum power example

V. Conclusion and Future works

In this paper, an algorithm is proposed that enables a parameter-dependent LP to be found whose solution approximates some nonlinear map for which data are available for the sake of identification. Two examples are given to illustrate the effectiveness of the proposed approach. Note that the proposed algorithm can be used to build a piece-wise LP identified models which enables to considerably extend the class of nonlinear functions that can be tightly approximated.